

LISTING OF THE CLAIMS

A detailed listing of claims is presented below. Please amend currently amended claims as indicated below including substituting clean versions for pending claims with the same number. In addition, clean text versions of pending claims not being currently amended that are under examination are also presented. It is understood that any claim presented in a clean version below has not been changed relative to the immediate prior version.

1. (Currently Amended) A method of transferring between types of conversion processes in a computer which converts instructions from a target instruction set to a host instruction set comprising the steps of:

executing code morphing software including an interpreter and a translator to generate host instructions from target instructions[[,]];

detecting at intervals whether the interpreter or the translator is operating for all instruction sequences[[,]];

increasing a count if the interpreter is operating and decreasing the count if the translator is operating[[,]];

changing from interpreting to translating a sequence of target instructions in response to the count reaching a selected maximum.

2. (Original) A method as claimed in Claim 1 in which the interval is a selected time period.

3. (Original) A method as claimed in Claim 1 in which the interval is a selected number of executed target instructions.

4. (Original) A method as claimed in Claim 1 in which the amount the count is increased at a detection of interpretation is selectable.

5. (Original) A method as claimed in Claim 1 in which the amount the count is decreased at a detection of translation is selectable.

6. (Original) A method as claimed in Claim 1 comprising the further steps of:

counting each instance in which a sequence of instructions is interpreted,

changing from interpreting to translating a sequence of target instructions when the count of instances reaches a selected maximum.

7. (Previously Presented) A method as claimed in Claim 1 comprising the further steps of:

gathering statistics regarding each sequence of instructions, and

optimizing translation of a sequence of instructions based on statistics gathered.

8. (Original) A method as claimed in Claim 1 in which the step of changing from interpreting to translating a sequence of target instructions when the count reaches a selected maximum includes translation with limited optimization, and

which further includes the steps of:

testing while executing a sequence of target instructions translated with limited optimization to determine whether the sequence should be further optimized, and

retranslating and further optimizing in response to the testing.

9. (Original) A method as claimed in Claim 8 in which the step of testing while executing a sequence of target instructions translated with limited optimization includes counting each instance in which a sequence of instructions is executed, and

the step of retranslating and further optimizing occurs when the count of instances reaches a selected maximum.

10. (Currently Amended) A method of optimizing execution by a computer which dynamically converts

instructions from a target instruction set to a host instruction set comprising the steps of:

providing a plurality of instruction conversion processes each providing a different level of optimization for converted instructions from a target instruction set to a host instruction set[[,]];

providing means for determining dynamically which conversion process to use to convert each sequence of instructions, said means depending on detecting at intervals which of the instruction conversion processes is operating for all instruction sequences[[,]]; and

converting a sequence of instructions using a conversion process determined by said means to convert the sequence of instructions.

11. (Original) A method as claimed in Claim 10 in which the conversion processes include interpretation and translation.

12. (Previously Presented) A method as claimed in Claim 10 in which the conversion processes include interpretation, translation with minimal optimization, translation with advanced optimization.

13. (Previously Presented) A method as claimed in Claim 10 in which the means for determining dynamically which conversion process to use to convert each sequence of

instructions further depends on the number of times each sequence is converted by a particular conversion process.

14. (Previously Presented) A method as claimed in Claim 10 in which the means for determining dynamically which conversion process to use to convert each sequence of instructions depends on a ratio of the number of times one conversion process is run compared to another conversion process.

15. (Previously Presented) A method as claimed in Claim 10 in which the means for determining dynamically which conversion process to use to convert each sequence of instructions

depends on the number of times each sequence is converted by a particular conversion process, and

depends on a ratio of the number of times one conversion process is run compared to another conversion process.

16. (Currently Amended) A method of transferring between types of conversion processes in a computer which converts instructions from a target instruction set to a host instruction set comprising the steps of:

executing code morphing software including an interpreter and a translator to generate host instructions from target instructions[[],];

comparing interpreter usage to translator usage when executing said code morphing software to produce an interpreter usage to translator usage factor, wherein said interpreter usage to translator usage factor is determined by detecting at intervals whether the interpreter or the translator is operating for all instruction sequences; and

changing from interpreting to translating a sequence of target instructions if the interpreter usage to translator usage factor crosses a threshold.

17. (Currently Amended) A method of transferring between types of conversion processes in a computer which converts instructions from a target instruction set to a host instruction set, said method comprising:

generating host instructions from target instructions by executing software comprising an interpreter and a translator;

during said generating, detecting at intervals whether said interpreter or said translator is operating for all instruction sequences;

in response to said detecting, increasing a count if the interpreter is operating and decreasing said count if said translator is operating;

in response to said count reaching a selected maximum, recording a sequence of instructions then being processed by said generating; and

changing from interpreting to translating said sequence of instructions upon said sequence of instructions next being encountered by said generating.

18. (Original) The method of Claim 17 wherein said interval is a selected time period.

19. (Original) The method of Claim 17 wherein said interval is a selected number of executed target instructions.

20. (Original) The method of Claim 17 wherein an amount of said increasing said count is selectable.

21. (Original) The method of Claim 17 wherein an amount of said decreasing said count is selectable.

22. (Original) The method of Claim 17 further comprising:

counting each instance in which a sequence of instructions is interpreted by said generating; and

changing from interpreting to translating said sequence of instructions when the count of instances reaches a selected maximum.

23. (Original) The method of Claim 22 further comprising:

gathering statistics regarding each sequence of
instructions; and

optimizing translation of a sequence of instructions
based on statistics gathered.